INFORMATION AND COMPUTATION **139**, 130–153 (1997)
ARTICLE NO. IC972667

Positive Tree Representations and Applications to

Symeon Bozapalidis

Department of Mathematics, Aristotle University of Thessaloniki, 54006 Thessaloniki, Greece

An effective classification of tree automata costed over the semirings $\mathbb{R}_+$ and $\mathbb{N}$ (and more generally of positive tree representations (PTR)) is achieved by means of a global behavior theory. Reducibility and minimality of PTRs is also investigated.

INTRODUCTION

In general, when weights are put on the transitions of a machine $\mathscr{M}$, a cost is attributed to any work this machine carries out, provided a final weight distribution ζ on the states of $\mathscr{M}$ is given.

There are two structurally different ways to compare machines. In the first method, the local, two machines, $\mathscr{M}$ and $\mathscr{M}'$, are given together with final distributions ζ and ζ', respectively, and we search to decide whether the cost of their functioning is the same. This is the known equality theorem established in [6] for K-Σ-automata and in [4] for K-Σ-algebras.

The second machine comparison method is global and is defined as follows: for each weighted machine $\mathscr{M}$ let $B(\mathscr{M})$ be the set of behaviors of $\mathscr{M}$ for all possible final distributions ζ. We say that $\mathscr{M}$ covers (or is equivalent to) $\mathscr{M}'$ whenever $B(\mathscr{M}) \supseteq B(\mathscr{M}')$ (or $B(\mathscr{M}) = B(\mathscr{M}')$). The notion of covering is widely used in theoretical computer science to classify various objects (see [7] for monoids of transformations, [10] for stochastic sequential machines, [2] for coalgebras, etc.).

One main goal of the present paper is to attack the last equivalence problem for positively weighted (bottom up) tree automata. Precisely such a machine is a triple $A = (\Sigma, Q, \alpha)$ formed by an input ranked alphabet Σ, a finite set Q of states, and a family of functions

$$\alpha_\sigma : Q^n \to K^Q, \qquad \sigma \in \Sigma_n, \qquad n \geqslant 0$$

describing the moves of A. (K denotes any of the semiring $\mathbb{R}_+$ of nonnegative reals or $\mathbb{N}$ of naturals.) The number $\alpha_\sigma(q_1, ..., q_n)(q)$ is the cost of the move $a_1 \cdots q_n \xrightarrow{\sigma} q$.

0890-5401/97 $25.00

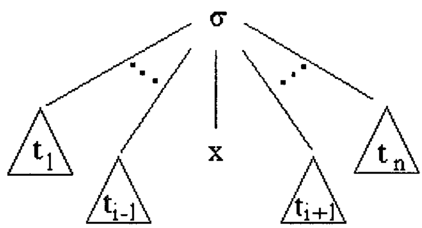

FIGURE 1

For each positive distribution ζ ($\zeta \in \mathbb{K}^{\text{card } Q}$) the corresponding behavior A_ζ: $T_\Sigma \to \mathbb{K}$ is given by

$$A_\zeta(t) = H_A(t) \cdot \zeta, \qquad t \in T_\Sigma,$$

where T_Σ stands for the set of all trees built up over Σ and $H_A\colon T_\Sigma \to \mathbb{K}^Q$ is the reachability map of A.

Then A covers A' whenever

$$\{A_\zeta / \zeta \in K^{\text{card } Q}\} \supseteq \{A'_\eta / \eta \in K^{\text{card } Q'}\},$$

that is, whenever for each positive distribution η on the states of A' we can determine a positive distribution ζ on the states of A such that $A_\zeta = A'_\eta$. It is shown that this comparison is effective.

In reality we solve this problem in the more general setting of positive tree representations. Tree representations with entries in an arbitrary semiring were introduced in [3] in order to characterize series on trees computed by tree modules, extending to trees the well-known word theorem of Salomaa and Soittola (cf. [11], Theorem 3.1).

Weighted word automata and matrix representations are equivalent notions but this is not the case for trees.

Intuitively, a positive tree representation (PTR) is a path-processing tree machine which consumes a tree independent of the path that follows.

The formal definitions pass through the monoid P_Σ of pruned trees which are catenations of terms of the form shown in Fig. 1 as well as the natural action $T_\Sigma \times R_\Sigma \to T_\Sigma$ depicted in Fig. 2. Thus a PTR is a finite set Q (states) together with two functions

$$y\colon \Sigma_o \to \mathbb{R}_+^{1\times n}, \qquad \varphi\colon P_\Sigma \to \mathbb{R}_+^{n\times n} \qquad (n = \text{card } Q)$$

(φ is a monoid morphism) which are mutually compatible in the sense that whenever a tree is factorized in two different ways (see Fig. 3) we require

$$y(c)\, \varphi(\tau) = y(c')\, \varphi(\tau').$$

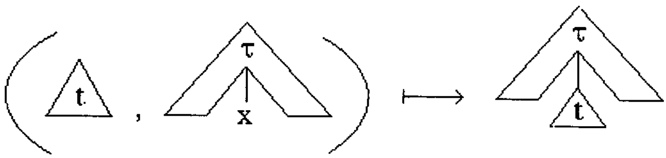

FIGURE 2

Therefore, a PTR $\Delta = (\Sigma, Q, y, \varphi)$ yields a function $\Delta\colon T_\Sigma \to \mathbb{R}_+^{1\times n}$ by

$$\Delta(t) = y(c)\,\varphi(\tau), \qquad t = c\tau \qquad (c \in \Sigma_o,\ \tau \in P_\Sigma).$$

A matrix L^Δ is associated with Δ as follows: we specify a linear, height increasing ordering on the trees of T_Σ and take the first linearly independent list $\Delta(t_1), \ldots, \Delta(t_\kappa)$ with the property that any other vector $\Delta(t)$, $t \in T_\Sigma$, is a linear combination of the vectors in the list. Then

$$L^\Delta = \begin{bmatrix} \Delta(t_1) \\ \vdots \\ \Delta(t_\kappa) \end{bmatrix}.$$

We must search the rows of L^Δ among the finite list $\Delta(t)$, height$(t) \leqslant$ card Q (Section 3). Call a PTR Δ minimal whenever the columns of L^Δ are positively independent (that is, no column of L^Δ is a positive combination of the other columns). Then it is shown that for each PTR Δ we can construct an equivalent minimal $\Delta_{\min}$.

The main result of the whole paper concerns covering characterization: for PTRs, Δ and Δ' the conditions below are mutually equivalent:

(i) $\Delta \geqslant \Delta'$.

(ii) There exists a matrix P with positive entries such that

$$\Delta(t) \cdot P = \Delta'(t) \qquad \text{for all} \quad t \in T_\Sigma.$$

(iii) There exists a matrix P with positive entries such that

$$\begin{cases} y(c) \cdot P = y'(c) & \text{for all} \quad c \in \Sigma_o \\ L^\Delta \cdot \varphi(\tau) \cdot P = L^\Delta \cdot P \cdot \varphi'(\tau) & \text{for all} \quad \tau \in \Sigma_o, \end{cases}$$

where P_n is the finite set of pruned trees $\tau = \tau_1 \cdots \tau_\kappa$, with $\kappa \leqslant n$, $|\tau_i| = 1$ for all i $(1 \leqslant i \leqslant \kappa)$ and height$(t) \leqslant n$, for all trees t that appear as a clone of a τ_i (see Fig. 4).

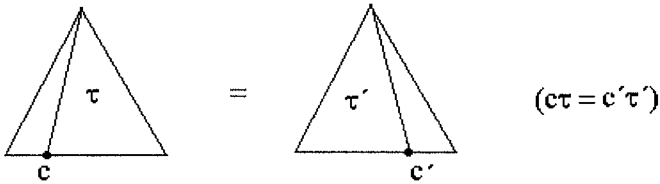

FIGURE 3

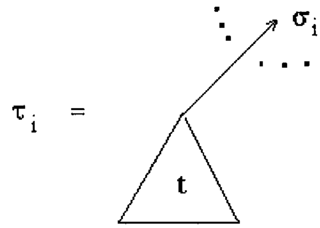

FIGURE 4

Applying this theorem to the PTR $\Delta(A)$ associated with a PTA A yields the next decidability result: Given PTAs A and A', we can decide whether or not $A \geqslant A'$.

1. BASIC TOPICS

A preliminary discussion is needed in order to fix our notation. We start with positiveness. A *positive set* is a subset A of an $\mathbb{R}$-vector space M such that

$$\alpha_i \in A \qquad \text{and} \qquad \lambda_i \geqslant 0 \quad (1 \leqslant i \leqslant n) \text{ implies } \sum_{i=1}^{n} \lambda_i \alpha_i \in A.$$

The smallest positive set including a subset $L \subseteq M$ is called a *positive hull* of L and is denoted ph(L); it holds that

$$\mathrm{ph}(L) = \left\{ \sum_{i=1}^{n} \lambda_i \alpha_i / \alpha_i \in L, \lambda_i \geqslant 0, n \geqslant 1 \right\},$$

i.e., ph(L) is the set of all *positive combinations* of the elements of L.

A sequence of points $x_1, ..., x_n$ of the space M is said to be *positively independent* if

$$x_i \notin \mathrm{ph}\{x_1, ..., x_{i-1}, x_{i+1}, ..., x_n\} \qquad \text{for all } i.$$

Next, consider a ranked alphabet $\mathbf{\Sigma} = \mathbf{\Sigma}_o \cup \mathbf{\Sigma}_1 \cup \cdots \cup \mathbf{\Sigma}_\kappa$. The set $T_\Sigma(x)$ of trees over Σ indexed by the variable $x \notin \Sigma$ is the smallest language over the alphabet $\Sigma \cup \{x\} \cup \{(\, , \, , \,)\}$ such that

(i) $\Sigma_o \cup \{x\} \subseteq T_\Sigma(x)$

(ii) if $\sigma \in \Sigma_n$ and $t_1, ..., t_n \in T_\Sigma(x)$, then $\sigma(t_1, ..., t_n) \in T_\Sigma(x)$.

$T_\Sigma(x)$ is converted into a monoid via the substitution operation: for $\tau, \pi \in T_\Sigma(x)$, $\tau\pi$ is the result of substituting τ at x inside π. Two subsets of $T_\Sigma(\kappa)$ are of interest:

— the set T_Σ of ground trees ($t \in T_\Sigma$ iff x does not appear on t) and

— the set P_Σ of pruned trees ($\tau \in P_\Sigma$ iff x occurs exactly once on t).

EXAMPLE 1. Take $\Sigma_o = \{a\}$, $\Sigma_2 = \{\sigma\}$; the trees in Fig. 5 belong to $T_\Sigma(x)$. The tree shown in Fig. 5b is ground (no appearance of x), whereas the tree shown in Fig. 5c is pruned.

P_Σ is actually a free monoid generated by all the trees of the form

$$\tau = \sigma(t_1, ..., t_{i-1}, x, t_{i+1}, ..., t_n) \qquad \sigma \in \Sigma_n, \quad t_j \in T_\Sigma. \tag{1}$$

This means that each $\pi \in P_\Sigma$ can be uniquely factorized as a product of trees of the form (1)

$$\pi = \tau_1 ... \tau_\kappa,$$

the number κ of factors being the length of π (denoted $|\pi|$).

For instance tree (c) in Fig. 5 is equal to the product $\tau_1 \tau_2 \tau_3$, where τ_1, τ_2, τ_3 are in Fig. 6. P_Σ acts canonically on the right on T_Σ and each tree $t \in T_\Sigma$ is obviously written

$$t = c\tau \qquad (c \in \Sigma_o, \tau \in P_\Sigma)$$

(the elements of Σ_o are often called *leaves*).

Size and height are convenient ways to measure trees. They are functions

$$\text{size, height}: T_\Sigma \to \mathbb{N} \ \ (= \text{natural numbers})$$

inductively defined by

$\text{size}(c) = 1$, $\text{height}(c) = 0$, for all $c \in \Sigma_o$

— for all $t \in \sigma(t_1, ..., t_n)$, $\sigma \in \Sigma_n$, $t_j \in T_\Sigma$

$\text{size}(t) = 1 + \text{size}(t_1) + \cdots + \text{size}(t_n)$ and

$\text{height}(\sigma(t_1, ..., t_n)) = 1 + \max\{\text{height}(t_i)/1 \leqslant i \leqslant n\}$;

that is, size(t) is the number of symbols of Σ appearing in t and height(t) is the length of the longest path of t.

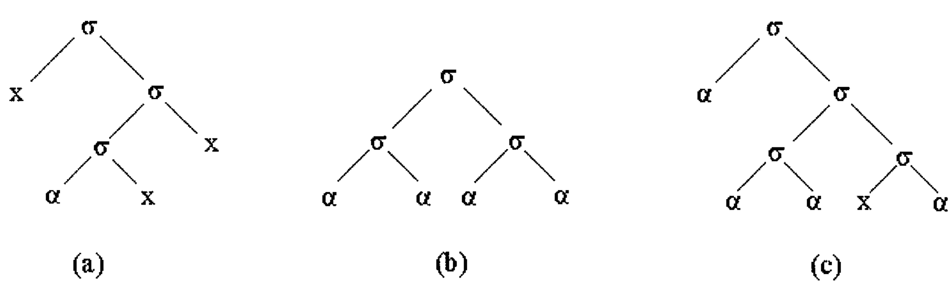

FIGURE 5

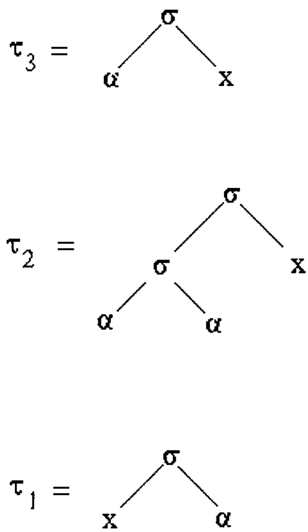

FIGURE 6

For decision questions attacked later on, we introduce P_n (n is a positive integer) to be the finite set of pruned trees τ whose length is $\leqslant n$,

$$\tau = \tau_1 \dots \tau_\kappa \qquad (\kappa \leqslant n)$$

$$\tau_j = \sigma_j(t_1^{(j)}, \dots, t_{i-1}^{(j)}, x, t_{i+1}^{(j)}, \dots, t_m^{(j)}) \qquad \sigma_j \in \Sigma_m \qquad 1 \leqslant j \leqslant \kappa,$$

and such that all trees $t_\lambda^{(j)}$ ($j = 1, \dots, \kappa$) have height $\leqslant n$. We refer to [4] for details.

2. POSITIVE TREE REPRESENTATIONS

Tree representations with entries in an arbitrary semiring have been investigated in [3]. The semiring $\mathbb{R}_+$ of nonnegative real numbers presents a special interest because this case gives new and deep results applicable to tree automata.

A *positive tree representation* is a 4-tuple $\Delta = (\Sigma, Q, y, \varphi)$, where Σ is a finite ranked alphabet (of inputs), Q is a finite set (of states), y is a function assigning a positive $1 \times n$ vector $y(c)$ to each leaf $c \in \Sigma_o$,

$$y\colon \Sigma_o \to \mathbb{R}_+^{1 \times n},$$

and φ is a function sending each tree $\tau \in P_\Sigma$ to an $n \times n$ matrix $\varphi(\tau)$ with positive entries

$$\varphi\colon P_\Sigma \to \mathbb{R}_+^{n \times n} \qquad (n = \text{card } Q).$$

We require φ to be a monoid morphism; i.e.,

$$\varphi(\tau_1 \tau_2) = \varphi(\tau_1)\, \varphi(\tau_2) \qquad \text{for all} \quad \tau_1, \tau_2 \in P_\Sigma.$$

In addition, y and φ are compatible in the sense that the condition

$$(*) \quad c\tau = c'\tau' \text{ implies } y(c)\,\varphi(\tau) = y(c')\,\varphi(\tau')$$

holds for all $c, c' \in \Sigma_o$ and $\tau, \tau' \in P_\Sigma$.

Interpretation. y is a system of initial *positive distributions* on the states of Δ. The transition of Δ is controlled by the matrices $\varphi(\tau)$, where $\varphi(\tau)_{ij}$ is the *positiveness* of the machine going to state j, given it had been in state i and fed with the tree τ. Condition $(*)$ expresses that Δ consumes a tree independent of the path it follows.

EXAMPLE 2. Let us specify an ordering on the symbols of our ranked alphabet $\Sigma = \{\sigma_1, ..., \sigma_n\}$ and for $s \in T_\Sigma \cup P_\Sigma$ denote by $\mathrm{size}_\sigma(s)$ the number of symbols $\sigma(\in \Sigma)$ occuring in s.

Consider the state set $Q = \{1, 2, ..., n\}$; then the function $\varphi: P_\Sigma \to \mathbb{R}_+^{n \times n}$,

$$\varphi(\tau) = \begin{pmatrix} \alpha^{\mathrm{size}_{\sigma_1}(\tau)} & & 0 \\ & \ddots & \\ 0 & & \alpha^{\mathrm{size}_{\sigma_n}(\tau)} \end{pmatrix} \qquad \alpha \in \mathbb{R}_+$$

is a monoid morphism compatible with $y: \Sigma_o \to \mathbb{R}_+^{1 \times n}$,

$$y(x) = (1, ..., 1, \alpha, 1, ..., 1),$$

where α is located at the place corresponding to c inside the ordered set Σ.

The above data are organized into a PTR Δ and the induced map $\Delta: T_\Sigma \to \mathbb{R}_+^{1 \times n}$ is

$$\Delta(t) = (\alpha^{\mathrm{size}_{\sigma_1}(t)}, ..., \alpha^{\mathrm{size}_{\sigma_n}(t)}), \qquad t \in T_\Sigma;$$

i.e., a Parikh-like function.

For any tree $T \in T_\Sigma$, a vector $\Delta(t) \in \mathbb{R}_+^{1 \times n}$ is defined by setting

$$\Delta(t) = y(c)\,\varphi(\tau) \qquad \text{if} \quad t = c\tau \ (c \in \Sigma_o, \tau \in P_\Sigma).$$

By $(*)$ above, Δ is well defined.

Fact 1. It holds that

$$\Delta(t\pi) = \Delta(t)\,\varphi(\pi) \qquad \text{for all} \quad t \in T_\Sigma, \pi \in P_\Sigma.$$

Indeed, if $t = c\tau$ $(c \in \Sigma_o,\ \tau \in P_\Sigma)$ then

$$\Delta(t\pi) = \Delta(c\tau\pi) = y(c)\,\varphi(\tau\pi) = y(c)\,\varphi(\tau)\,\varphi(\pi) = \Delta(t)\,\varphi(\pi).$$

Now, let η be a positive distribution over the states of the PTR Δ (that is, a vector $\eta \in \mathbb{R}_+^{n\times 1}$, $n = \text{card } Q$); then for any tree $t \in T_\Sigma$ and $\tau \in P_\Sigma$ we set

$$\Delta_\eta(t) = \Delta(t)\,\eta, \qquad \eta(\tau) = \varphi(\tau)\,\eta.$$

Δ_η is a function from T_Σ to $\mathbb{R}_+$, whereas $\eta(-)$ maps P_Σ into $\mathbb{R}_+^{1\times n}$.

Fact 2. For all $t \in T_\Sigma$, $\tau \in P_\Sigma$ we have

$$\Delta_\eta(t\tau) = \Delta(t)\,\eta(\tau).$$

The above two facts will be used repeatedly throughout this paper without specific mention.

Two positive tree representations Δ and Δ' with state sets Q and Q', respectively, are said to be state equivalent if there exist functions

$$\{1, 2, ..., \text{card } Q\} \underset{\theta'}{\overset{\theta}{\rightleftarrows}} \{1, 2, ..., \text{card } Q'\}$$

such that for all $t \in T_\Sigma$

$$\Delta(t)\cdot e_i = \Delta'(t)\cdot e_{\theta(i)} \qquad \text{and} \qquad \Delta'(t)\cdot e_\kappa = \Delta(t)\cdot e_{\theta'(\kappa)},$$

where e_i is the column vector having 1 in the ith place and 0 elsewhere.

Clearly, state equivalence is reflexive, symmetric, and transitive, i.e., an equivalence relation.

3. THE MATRIX L^Δ

Given a positive tree representation

$$\Delta = (\Sigma, Q, y, \varphi)$$

we symbolize by $L(\Delta)$ the infinite matrix

$$L(\Delta) = \begin{bmatrix} \Delta(t_1) \\ \Delta(t_2) \\ \vdots \end{bmatrix}$$

such that all the vectors of the form $\Delta(t)$ are listed above and their order is induced by some fixed linear order on the trees $t \in T_\Sigma$ respecting tree-height, i.e., such that

$$t \leqslant t' \text{ implies height}(t) \leqslant \text{height}(t').$$

Denote by $V_m(\Delta)$ the subspace of $\mathbb{R}^{1\times n}$ generated by the vectors $\Delta(t)$, height$(t)\leqslant m$. Then

$$V_o(\Delta)\subseteq V_1(\Delta)\subseteq \cdots \subseteq \mathbb{R}^{1\times n} \qquad n=\text{card } Q$$

hence,

$$1\leqslant \dim V_o(\Delta)\leqslant \dim V_1(\Delta)\leqslant \cdots \leqslant n.$$

Therefore for some index $i\leqslant n-1$

$$V_i(\Delta)=V_{i+1}(\Delta).$$

Then $V_{i+1}(\Delta)=V_{i+2}(\Delta)$. To prove this assertion we observe that

$$\delta\in V_{i+2}(\Delta) \text{ implies } \delta=\sum_j \lambda_j\Delta(t_j) \text{ height}(t_j)\leqslant i+2.$$

The case height$(t_j)\leqslant i+1$ is clear so that it remains to study the case height$(t_j)=i+2$. For all such j we have

$$t_j=s_j\pi_j, \text{ height}(s_j)=i+1, \pi_j\in P_\Sigma, |\pi_j|=1.$$

Hence,

$$\Delta(s_j)\in V_{i+1}(\Delta)=V_i(\Delta);$$

that is,

$$\Delta(s_j)=\sum_\kappa \alpha_{\kappa j}\Delta(u_\kappa) \qquad \text{height}(u_\kappa)\leqslant i, \qquad \alpha_{k_j}\in\mathbb{R}.$$

Finally,

$$\delta=\sum_j \lambda_j\Delta(t_j)=\sum_j \lambda_j\Delta(s_j\pi_j) \overset{\text{fact 1}}{=} \sum_j \lambda_j\Delta(s_j)\,\varphi(\pi_j)$$

$$\sum_j \lambda_j\left(\sum_\kappa \alpha_{\kappa j}\Delta(u_\kappa)\right)\varphi(\pi_j)=\sum_{j,\kappa}\lambda_j\alpha_{\kappa j}(u_\kappa)\,\varphi(\pi_j) \overset{\text{fact 1}}{=} \sum_{j,\kappa}\lambda_j\alpha_{\kappa j}\Delta(u_\kappa\pi_j).$$

For any j, κ either the height of $u_\kappa\pi_j$ is at most $i+1$, or $u_\kappa\pi_j$ has fewer paths of length $i+2$ than $t_j=s_j\cdot\pi_j$; in the last case we repeat the procedure above and finally we get $\delta\in V_{i+1}(\Delta)$.

An induction on p shows that $V_i(\Delta)=V_{i+p}(\Delta)$ for $p=1, 2, ...,$ so that all the rows of $L(\Delta)$ belong to the subspace $V_{n-1}(\Delta)$, $n=\text{card } Q$.

Now assume that

$$\delta_1, \delta_2, ..., \delta_m \qquad m\leqslant \text{card } Q$$

are the first rows in $L(\Delta)$ (in the order of vectors in it) which constitute a basis of the vector space

$$\bigcup_{\kappa=0}^{\infty} V_\kappa(\Delta).$$

The matrix L^Δ is defined by

$$L^\Delta = \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_m \end{bmatrix}.$$

The rank of L^Δ is by definition the rank of the representation Δ.

EXAMPLE 3. Let us consider the function eval: $T_\Sigma \to \mathbb{R}_+$ evaluating arithmetic expressions comprising addition and multiplication (i.e., Σ is the ranked alphabet $\Sigma_1 = \{+, \cdot\}$, $\Sigma_o = \{\kappa_1, ..., \kappa_\lambda\}$ with $\kappa_j \in \mathbb{R}_+$).

Next the data $\varphi: P_\Sigma \to \mathbb{R}_+^{2\times 2}$, $y: \Sigma_o \to \mathbb{R}_+^{1\times 2}$

$$\varphi\left(\begin{array}{ccc} & + & \\ x \diagup & & \diagdown t \end{array}\right) = \begin{pmatrix} 1 & \mathrm{eval}(t) \\ 0 & 1 \end{pmatrix} = \varphi\left(\begin{array}{ccc} & + & \\ t \diagup & & \diagdown x \end{array}\right)$$

$$\varphi\left(\begin{array}{ccc} & \bullet & \\ x \diagup & & \diagdown t \end{array}\right) = \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{eval}(t) \end{pmatrix} = \varphi\left(\begin{array}{ccc} & \bullet & \\ t \diagup & & \diagdown t \end{array}\right)$$

$$y(\kappa_i) = (1 \quad \kappa_i), \qquad 1 \leqslant i \leqslant \lambda$$

constitute a positive tree representation Δ. Its matrix is of the form

$$L^\Delta = \begin{pmatrix} 1 & \kappa \\ 1 & \mu \end{pmatrix} \qquad \kappa \neq \mu.$$

4. THE SUBSTITUTION LEMMA

Let

$$\Delta = (\Sigma, Q.y, \varphi)$$

be a PTR; since P_Σ is the free monoid generated by the trees of the form

$$\pi = \sigma(t_1, ..., t_{i-1}, x, t_{i+1}, ..., t_n),$$

the monoid morphism φ is uniquely determined by its values on these trees.

Let a_i be the ith column of a matrix $\varphi(\pi)$, $\pi \in P_\Sigma$, $|\pi| = 1$. Let, further, a be a positive column vector such that

$$L^\Delta.a = L^\Delta.a_i$$

and denote by Δ' the PTR derived from Δ by replacing the column a_i with a into $\varphi(\pi)$. Then

SUBSTITUTION LEMMA. *Δ' is state equivalent to Δ.*

Proof. We first need to show that Δ' is really a PTR; that is, we must establish the coherence axiom (*). For this let $c \in \Sigma_o$ and $\tau \in P_\Sigma$. We shall show that

$$y'(c)\,\varphi'(\tau) = \Delta(c\tau).$$

If π does not appear inside τ then we are done.

Assume now that π appears inside τ and let

$$\tau = \pi_1 \pi \pi_2, \qquad \text{with} \quad \pi_1, \pi_2 \in P_\Sigma \quad \text{and} \quad \pi_1 \ \pi\text{-free}.$$

Then

$$\begin{aligned} y'(c)\,\varphi'(\tau) &= y(c)\,\varphi'(\pi_1 \pi \pi_2) = y(c)\,\varphi'(\pi_1)\,\varphi'(\pi)\,\varphi'(\pi_2) \\ &= y(c)\,\varphi(\pi_1)\,\varphi'(\pi)\,\varphi'(\pi_2) = \Delta(c\pi_1)\,\varphi'(\pi)\,\varphi'(\pi_2). \end{aligned}$$

By hypothesis $L^\Delta.a = L^\Delta.a_i$ and since $\Delta(c\pi_1)$ is written as a linear combination of the rows of L^Δ, we get

$$\Delta(c\pi_1)\,\varphi'(\pi) = \Delta(c\pi_1)\,\varphi(\pi) = \Delta(c\pi_1\pi).$$

As π_2 has fewer occurences of π than τ, an induction argument can be applied to show that

$$y'(c)\,\varphi'(\tau) = \Delta(c\pi_1\tau)\,\varphi'(\pi_2) = \Delta(c\tau_1\pi\pi_2) = \Delta(c\tau)$$

as claimed.

Assume next

$$c\tau = c'\tau' \qquad \text{with} \quad c, c' \in \Sigma_o \quad \text{and} \quad \tau, \tau' \in P_\Sigma.$$

Then

$$y'(c)\,\varphi'(\tau) = \Delta(c\tau) = \Delta(c'\tau') = y'(c')\,\varphi'(\tau').$$

In order to show that Δ' is state-equivalent to Δ it suffices to show that

$$\Delta'(t) = \Delta(t) \qquad \text{for all} \quad t \in T_\Sigma$$

which immediately comes from the previous discussion:

$$\Delta'(t) = y'(c)\,\varphi'(\tau) = \Delta(c\tau) = \Delta(t). \quad \blacksquare$$

Call a tree representation Δ *reduced* if no two columns of the matrix L^{Δ} are identical.

PROPOSITION 1. *Every PTR $\Delta = (\Sigma, Q, y, \varphi)$ is state equivalent to a reduced one.*

Proof. Denote by $\omega_1, ..., \omega_n$ the columns of L^{Δ} and assume that $\omega_i = \omega_j$ $(i < j)$. Let $a = (a_1, ..., a_n)^{\mathrm{T}}$ be a column of a matrix of the form $\varphi(\tau)$, $|\tau| = 1$, and let $\bar{a}$ be the column defined by

$$\bar{a}_i = 0, \quad \bar{a}_j = a_j + a_i, \quad \bar{a}_\kappa = a_\kappa \qquad \kappa \neq i, j.$$

Then

$$L^{\Delta} \cdot a = \sum_{\kappa=1}^{n} a_\kappa \omega_\kappa = a_1 \omega_1 + \cdots + 0 \cdot \omega_i + \cdots + (a_j + a_i)\, \omega_j + \cdots + a_n \omega_n = L^{\Delta} \cdot \bar{a}.$$

Replacing a by $\bar{a}$ we get a state equivalent tree representation Δ' (see the substitution lemma). Repeating this construction we arrive at a tree representation $\Delta'' = (\Sigma, Q, y'', \varphi'')$ such that the ith row of all matrices $\varphi''(\tau)$, $\tau \in P_\Sigma$, is the zero row, whereas for all $c \in \Sigma_o$, $g'(c)$ is defined to be 0 at the ith entry and equal to $y(c)_\kappa$ at the remainder entries $(\kappa \neq i)$.

Let Δ''' be the positive tree representation resulting from Δ'' by deleting ith row and ith column from all matrices $\varphi''(\tau)$ as well as the ith entry from all $y''(c)$. Then Δ''' has $n-1$ states and is clearly state equivalent to Δ: this equivalence is realized by the pair of functions

$$\{1, 2, ..., \operatorname{card} Q\} \underset{\theta'}{\overset{\theta}{\rightleftarrows}} \{1, 2, ..., i-1, i+1, ..., \operatorname{card} Q'\},$$

where θ' is the obvious inclusion and

$$\begin{aligned} \theta(\lambda) &= \lambda, && \text{if} \quad \lambda \neq i \\ &= j, && \text{if} \quad \lambda = i. \end{aligned}$$

If $L^{\Delta'''}$ continues to have two identical columns we repeat the procedure. ∎

EXAMPLE 4. Consider a finite ranked alphabet Σ and the PTR $\Delta = (\Sigma, Q, y, \varphi)$ with

$$Q = \{1, 2, 3\}$$

$$y(c) = (1 \quad 1) \qquad \text{for all} \quad c \in \Sigma_o$$

$$\varphi(\tau) = \begin{pmatrix} 1 & \operatorname{size}(\tau) & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \text{for all} \quad \tau \in P_\Sigma,$$

where size(τ) is the number of symbols of Σ occuring in τ. For all $t \in T_\Sigma$ it holds

$$\Delta(t) = (1 \quad \mathrm{size}(t) \quad 1).$$

Thus,

$$L^\Delta = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \kappa & 1 \end{pmatrix}, \qquad \kappa \text{ a fixed positive integer} > 1.$$

The first and third columns are equal, so an application of the procedure displayed above yields the reduced 2-state representation Δ' with

$$c \mapsto (1 \quad 1), \qquad \tau \mapsto \begin{pmatrix} 1 & \mathrm{size}(\tau) \\ 0 & 1 \end{pmatrix}.$$

4. REPRESENTATION COVERING

Covering is a convenient way to compare objects in theoretical computer science; the nature of the examined objects determines the type of covering we adopt (cf. [2, 7, 10]).

In Theorem 2 below, we show that PTR covering is finitely checkable.

The *behavior set* of a positive tree representation

$$\Delta = (\Sigma, Q, y, \varphi)$$

is by definition

$$B(\Delta) = \{\Delta_\zeta / \zeta \in \mathbb{R}_+^{n \times 1}, n = \mathrm{card}\, Q\}$$

(recall that $\Delta_\zeta \colon T_\Sigma \to \mathbb{R}_+$ is given by $\Delta_\zeta(t) = \Delta(t) \cdot \zeta$).

Fact 3. $B(\Delta)$ is the positive hull of the set of functions

$$B_e(\Delta) = \{\Delta_{e_1}, ..., \Delta_{e_n}\},$$

where $\{e_1, ..., e_n\}$ is the standard basis of $\mathbb{R}^n$ and n is the number of states of Δ.

Indeed, for each $\Delta_\zeta \in B(\Delta)$ and all $t \in T_\Sigma$ we have

$$\Delta_\zeta(t) = \Delta(t)\, \zeta = \sum_{\kappa=1}^{n} \zeta_\kappa \Delta_{e_\kappa}(t);$$

i.e., $\Delta_\zeta = \sum_{\kappa=1}^{n} \zeta_\kappa \Delta_{e_\kappa}$, $\zeta_\kappa \in \mathbb{R}_+$ ($\kappa = 1, 2, ..., n$).

We immediately deduce that Δ and Δ' are equivalent iff

$$B_e(\Delta) \subseteq B(\Delta) \qquad \text{and} \qquad B_e(\Delta') \subseteq B(\Delta).$$

On the other hand, state equivalence implies equivalence:

$$B_e(\Delta) = B_e(\Delta') \Rightarrow B(\Delta) = \text{ph } B_e(\Delta) = \text{ph } B_e(\Delta') = B(\Delta').$$

Let Δ, Δ' be two PTRs. We say that Δ *covers* Δ' (notation $\Delta \geqslant \Delta'$) whenever $B(\Delta) \supseteq B(\Delta')$. Δ and Δ' are termed *equivalent* if $B(\Delta) = B(\Delta')$. The next result is crucial.

THEOREM 1. *Let $\Delta = (\Sigma, Q, y, \varphi)$ be an n-state PTR such that some column of L^Δ is a positive combination of the other columns. Then Δ is equivalent to an $(n-1)$-state PTR Δ'.*

Proof. Let $\omega_1, ..., \omega_n$ be the columns of L^Δ and assume that ω_i is written as a positive combination of $\omega_1, ..., \omega_{i-1}, \omega_{i+1}, ..., \omega_n$.

Let $a = (a_1, ..., a_n)^{\mathrm{T}}$ be a (nonzero) column of a matrix $\varphi(\tau)$, $|\tau| = 1$. Then

$$L^\Delta \cdot a = \sum_{i+1}^{a} a_i \omega_i \in \text{ph}\{\omega_1, ..., \omega_n\} = \text{ph}\{\omega_1, ..., \omega_{i-1}, \omega_{i+1}, ..., \omega_n\};$$

that is, there exists a vector $\bar{a} \in \mathbb{R}_+^{1\times n}$ with $\bar{a}_i = 0$ such that

$$L^\Delta \cdot a = \sum_{j \neq i} \bar{a}_j \omega_j = L^\Delta \cdot \bar{a}.$$

By the substitution lemma, from Δ we get a state-equivalent (and thus an equivalent) PTR Δ' having the property that the ith row of all matrices $\varphi(\tau)$ is the zero row.

Take $\xi \in \mathbb{R}_+^n$ $(n = \text{card } Q)$; arguing as above we can determine $\bar{\xi} \in \mathbb{R}_+^n$ with $\bar{\xi}_i = 0$ and

$$\Delta'(t)\, \xi = \Delta'(t)\, \bar{\xi} \qquad \text{for all} \quad t \in T_\Sigma.$$

Now, deleting the ith state from Δ', we get an equivalent PTR Δ'' having $(n-1)$ states. By transitivity of equivalence, Δ'' is equivalent to the initial PTR Δ, and the proof is completed. ∎

A PTR Δ is *minimal* if the set of column vectors of L^Δ is positively independent.

COROLLARY. *For each PTR Δ we can construct an equivalent minimal PTR $\Delta_{\min}$.*

Remark. The previous result states that PTRs (i.e., path processing tree machines) form a good class of machines since the minimization principle is applicable on them.

The main result of the paper is the following:

THEOREM 2. *The conditions below are mutually equivalent*:

(i) $\Delta \geqslant \Delta'$

(ii) *there exists a positive matrix* P (*i.e., all the entries of* P *are nonnegative real numbers*) *such that*

$$\Delta(t) \cdot P = \Delta'(t) \qquad \text{for all} \quad t \in T_\Sigma$$

(iii) *there exists a positive matrix* P *such that*

$$\begin{cases} y(c) \cdot P = y'(c), & \text{for all} \quad c \in \Sigma_o \\ L^\Delta \cdot \varphi(\tau) \cdot P = L^\Delta \cdot P \cdot \varphi'(\tau), & \text{for all} \quad \tau \in P_\Sigma \end{cases}$$

(iv) *there exists a positive matrix* P *such that*

$$\begin{cases} y(c) \cdot P = y'(c), & \text{for all} \quad c \in \Sigma_o \\ L^\Delta \cdot \varphi(\tau) \cdot P = L^\Delta \cdot P \cdot \varphi'(\tau), & \text{for all} \quad \tau \in P_n \end{cases}$$

Proof. Let e_i be the vector of $\mathbb{R}_+^{1 \times n}$ having 1 at the ith place and 0 elsewhere. If (i) holds true, there are $\zeta_\kappa \in \mathbb{R}_+^{1 \times n}$ ($\kappa = 1, ..., n$) such that

$$\Delta'_{e_\kappa} = \Delta_{\zeta_\kappa} \qquad \kappa = 1, 2, ..., n.$$

but then

$$\begin{aligned} \Delta'(t) &= [\Delta'_{e_1}(t) \cdots \Delta'_{e_n}(t)] \\ &= [\Delta_{\zeta_1}(t) \cdots \Delta_{\zeta_n}(t)] \\ &= [\Delta'(t)\, \zeta_1 \cdots \Delta(t)\, \zeta_n] \\ &= \Delta(t)[\zeta_1 \cdots \zeta_n] \\ &= \Delta(t) \cdot P, \end{aligned}$$

where P is the matrix formed by catenating the vectors $\zeta_1, ..., \zeta_n$ taken in this order.

Assume, next, that (ii) holds and let $\Delta'_\zeta \in B(\Delta')$; then for $\xi = P\zeta$ we have

$$\Delta_\xi(t) = \Delta_{P\zeta}(t) = \Delta(t)\, P\zeta = \Delta'(t)\, \zeta = \Delta'_\zeta(t);$$

i.e., $\Delta_\xi = \Delta'_\zeta$ and this proves $\Delta'_\zeta \in B(\Delta)$, so that $B(\Delta') \subseteq B(\Delta)$ and thus $\Delta' \leqslant \Delta$.

The implication (ii) $\Rightarrow$ (iii) is rather easy: evaluating $\Delta(t) \cdot P = \Delta'(t)$ at $t = c \in \Sigma_o \subseteq T_\Sigma$ we get $y(c) \cdot P = y'(c)$. On the other hand, for any row $\Delta(t)$ of $L^\Delta (t \in T_\Sigma)$ and any $\tau \in P_\Sigma$ we have

$$\begin{aligned} \Delta(t)\, \varphi(\tau)\, P &= \Delta(t\tau)\, P = \Delta'(t\tau) \\ &= \Delta'(t)\, \varphi'(\tau) = \Delta(t)\, P\varphi'(\tau) \end{aligned}$$

and thus $L^\Delta \cdot \varphi(\tau) \cdot P = L^\Delta \cdot P \cdot \varphi'(\tau)$ as wanted.

(iii) $\Rightarrow$ (iv) trivial because $P_n \subseteq P_\Sigma$.

(iv) $\Rightarrow$ (iii) Let $c \in \Sigma_o$ and $\tau \in P_n$ and

$$y(c) = \sum_{j=1}^{m} \beta_j \Delta(t_j) \qquad \beta_j \in \mathbb{R},$$

where $\Delta(t_1), ..., \Delta(t_m)$ are the rows of L^Δ. Then

$$\begin{aligned} \Delta(c\tau)\, P = y(c)\, \varphi(\tau)\, P &= \sum_j \beta_j \Delta(t_j)\, \varphi(\tau)\, P \\ &= \sum_j \beta_j \Delta(t_j)\, P\varphi'(\tau) = y(c)\, P\varphi'(\tau) \\ &= y'(c)\, \varphi'(\tau) = \Delta'(c\tau). \end{aligned}$$

In other words, we have proved that

$$\Delta(t)\, P = \Delta'(t) \qquad \text{for all } t, \quad \text{height}(t) \leqslant n. \tag{$**$}$$

Next, let $\tau = \tau_1 \cdots \tau_\kappa$, $\kappa > n$, $|\tau_i| = 1$ for all i and take row $\Delta(t)$ of L^Δ; then the vectors

$$\Delta(t), \Delta(t\tau_1), ..., \Delta(t\tau_1 \cdots \tau_\kappa)$$

must be linearly dependent and as $\Delta(t) \neq 0$, there exists an index $\lambda < \kappa$ such that

$$\Delta(t\tau_1 \cdots \tau_\lambda) = \sum_{i<\lambda} \alpha_i \Delta(t\tau_1 \cdots \tau_i) \qquad \alpha_j \in \mathbb{R}.$$

Multiplying on the right both sides above by the matrix $\varphi(\tau_{\lambda+1} \cdots \tau_\kappa)$ we get

$$\Delta(t\tau) = \Delta(t\tau_1 \cdots \tau_\lambda \tau_{\lambda+1} \cdots \tau_\kappa) = \sum_{i<\lambda} \alpha_i \Delta(t\tau_1 \cdots \tau_i \tau_{\lambda+1} \cdots \tau_\kappa).$$

We put

$$\bar{\tau} = \tau_1 \cdots \tau_i \tau_{\lambda+1} \cdots \tau_\kappa \qquad i = 1, 2, ..., \lambda - 1$$

and write

$$t\bar{\tau}_i = s\pi_i \qquad s \in T_\Sigma, \quad \pi_i \in P_\Sigma.$$

Let us decompose $\Delta(s)$ along the rows of L^Δ (see Fig. 7)

$$\Delta(s) = \sum_j \mu_j \Delta(t_j) \qquad \mu_j \in \mathbb{R}.$$

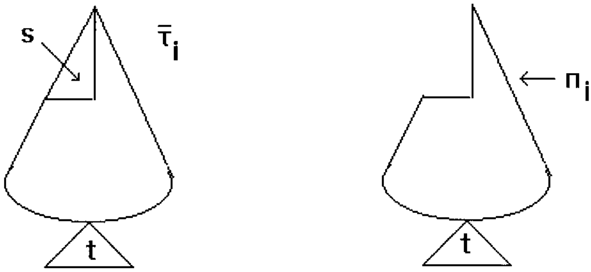

FIGURE 7

Then

$$
\begin{aligned}
\Delta(t)\,\varphi(\tau)\,P &= \sum_i \alpha_i \Delta(t)\,\varphi(\bar{\tau}_i)\,P \\
&= \sum_i \alpha_i \Delta(s)\,\varphi(\pi_i) \\
&= \sum_i \alpha_i \sum_j \mu_j \Delta(t_j)\,\varphi(\pi_i)\,P \\
&= \sum_i \alpha_i \sum_j \mu_j \Delta(t_j \pi_i)\,P \\
&= \sum_i \alpha_i \sum_j \mu_j \Delta(t)\,\varphi(\omega_{ji})\,P \\
&= \sum_i \alpha_i \sum_j \mu_j \Delta(t)\,P\varphi'(\omega_{ji}) \\
&= \sum_i \alpha_i \sum_j \mu_j \Delta'(t)\,P\varphi'(\omega_{ji}) \\
&= \Delta'(t\tau) = \Delta'(t)\,\varphi'(\tau) = \Delta(t)\cdot P\cdot \varphi'(\tau),
\end{aligned}
$$

where ω_{ji} is the unique tree of P_Σ satisfying the equality $t_j \pi_i = \omega_{ji}$ (see Fig. 8).

Since $\Delta(t)$ is an arbitrary row of L^Δ we have proved $L^\Delta \cdot \varphi(\tau)\,P = L^\Delta P \cdot \varphi'(\tau)$, for all $\tau \in T_\Sigma$.

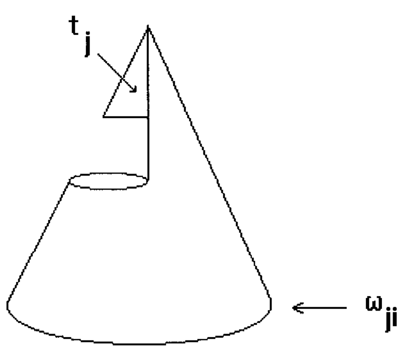

FIGURE 8

Finally, the implication (iii) $\Rightarrow$ (ii) comes arguing as at the beginning of (iv) $\Rightarrow$ (iii). ∎

Remark. From Theorem 2(ii), we get

$$\Delta \geqslant \Delta' \Rightarrow \operatorname{rank}(\Delta) \geqslant \operatorname{rank}(\Delta').$$

PROPOSITION 2. *Assume that* $\Delta \geqslant \Delta'$ *and* $\operatorname{rank}(\Delta) = \operatorname{rank}(\Delta')$. *Then there is a positive matrix* P *such that* $L^{\Delta} \cdot P = L^{\Delta}$.

Proof. Suppose L^{Δ} to be constructed by the rows $\Delta(t_1), ..., \Delta(t_m)$ and denote $J(\Delta, \Delta')$ the matrix constructed by the corresponding rows $\Delta'(t_1), ..., \Delta'(t_m)$, then

$$\Delta'(t_j) = \Delta(t_j) \cdot P \qquad j = 1, ..., m,$$

where P is the positive matrix coming from Theorem 2(ii). For any $t \in T_{\Sigma}$, let

$$\Delta(t) = \sum_j \lambda_j \Delta(t_j) \qquad \lambda_j \in \mathbb{R}$$

then

$$\Delta(t) = \Delta(t) \cdot P = \sum_j \lambda_j \Delta(t_j) \cdot P = \sum_j \lambda_j \Delta'(t_j).$$

This means that

$$\operatorname{rank} J(\Delta, \Delta') = \operatorname{rank}(\Delta').$$

We assert that the rows of $L^{\Delta'}$ are a subset of those of $J(\Delta, \Delta')$. If this is not true, let $\Delta'(t)$ be the first row of $L^{\Delta'}$ which is not a row of $J(\Delta, \Delta')$. Then the row $\Delta(t)$ is not in L^{Δ}, hence it is a linear combination of the rows of L^{Δ} preceding $\Delta(t)$ (in the initially fixed order). This would imply that $\Delta'(t)$ is a linear combination of other rows of $L^{\Delta'}$, a contradiction. Thus the rows of $L^{\Delta'}$ are a subset of the rows of $J(\Delta, \Delta')$. As $\operatorname{rank}(\Delta) = \operatorname{rank} J(\Delta, \Delta') = \operatorname{rank}(\Delta')$, we have $L^{\Delta'} = J(\Delta, \Delta')$ and this achieves the proof. ∎

THEOREM 3. *Assume that* Δ *and* Δ' *are equivalent PTRs with* n *and* n' *states respectively. Then*

(i) $\operatorname{rank}(\Delta) = \operatorname{rank}(\Delta')$

(ii) $\operatorname{ph}\{\omega_1^{\Delta}, ..., \omega_n^{\Delta}\} = \operatorname{ph}\{\omega_1^{\Delta'}, ..., \omega_n^{\Delta'}\}$,

where ω_i^{Δ} (*respectively* $\omega_i^{\Delta'}$) *denotes the* i *th column of the matrix* L^{Δ} (*respectively of* $L^{\Delta'}$).

Proof. We have

$$\Delta \text{ equivalent to } \Delta$$

iff

$$\Delta \geqslant \Delta' \qquad \text{and} \qquad \Delta' \geqslant \Delta$$

which implies

$$\operatorname{rank}(\Delta) \geqslant \operatorname{rank}(\Delta') \geqslant \operatorname{rank}(\Delta)$$

hence assertion (i).

From the previous proposition we get the equalities $L^{\Delta'} = L^{\Delta} \cdot P$ and $L^{\Delta} = L^{\Delta'} \cdot P'$ which mean that any column of $L^{\Delta'}$ is a positive combination of the columns of $L^{\Delta'}$ and vice versa, hence (ii). ∎

PROPOSITION 4. *Let us keep the notations of the previous theorem. Then for state equivalent PTRs Δ and Δ' with n and n' states respectively, it holds*

$$\{\omega_1^{\Delta}, ..., \omega_n^{\Delta}\} = \{\omega_1^{\Delta'}, ..., \omega_{n'}^{\Delta'}\}.$$

If, further, Δ and Δ' are reduced, then $n = n'$ and the columns of L^{Δ} are a permutation of the columns of $L^{\Delta'}$.

Proof. State equivalence implies equivalence, so that if

$$L^{\Delta} = \begin{bmatrix} \Delta(t_1) \\ \vdots \\ \Delta(t_k) \end{bmatrix}$$

then

$$L^{\Delta'} = \begin{bmatrix} \Delta'(t_1) \\ \vdots \\ \Delta'(t_k) \end{bmatrix}.$$

Next, let

$$\omega_i^{\Delta} = \begin{bmatrix} \Delta_{e_i}(t_1) \\ \vdots \\ \Delta_{e_i}(t_k) \end{bmatrix}$$

be the ith column of L^{Δ}; since by hypothesis Δ and Δ' are state equivalent, we have $\Delta_{e_i} = \Delta'_{e_i}$ for some index j $(1 \leqslant j \leqslant n')$. Consequently, $\omega_i^{\Delta} = \omega_j^{\Delta'}$ and our assertion will be completely established by reversing the arguments.

Finally, if both Δ and Δ' are reduced, then no two columns of L^{Δ} and no two columns of $L^{\Delta'}$ are identical, and the conclusion comes directly from the above facts. ∎

5. APPLICATION TO POSITIVE TREE AUTOMATA

A *positive tree automaton* (PTA) is a structure $A=(\Sigma, Q, \alpha)$ consisting of a finite ranked alphabet Σ of inputs, a finite set Q of states, and a family of functions of the form

$$\alpha_\sigma : Q^n \to \mathbb{R}_+^Q \qquad \sigma \in \Sigma_n, n \geqslant 0$$

that describe the moves of A; in particular, for every $c \in \Sigma_o$, α_c is a function from Q to $\mathbb{R}_+$. The number $\alpha_\sigma(q_1, ..., q_n)(q)$ expresses the positiveness of A going to the state q, given it had been in the (vector) state $(q_1, ..., q_n)$ and fed with the symbol σ (i.e., it is the cost of the move $q_1 \cdots q_n \xrightarrow{\sigma} q$).

The *reachability map* of A is the function $H_A : T_\Sigma \to \mathbb{R}_+^Q$ inductively defined by

— $H_A(c) = \alpha_c$, for all $c \in \Sigma_o$

— $H_A(\sigma(t_1, ..., t_n)) = \bar{\alpha}_\sigma(H_A(t_1), ..., H_A(t_n))$, for all $\sigma \in \Sigma_n$, $t_j \in T_\Sigma$, where

$$\bar{\alpha}_\sigma : (\mathbb{R}_+^Q)^n \to \mathbb{R}_+^Q$$

is the "multipositive" extension of α_σ; i.e.,

$$\bar{\alpha}_\sigma(x_1, ..., x_n) = \sum x_1(q_1) \cdots x_n(q_n)\, \alpha_\sigma(q_1, ..., q_n) \qquad \text{for all} \quad x_1, ..., x_n \in \mathbb{R}_+^Q$$

the sum running through $q_1, ..., q_n \in Q$.

Remark. Tree automata with costs over a field were introduced by Berstel and Reutenauer [1] (where a slight different formalism is used). Additional material on this subject can be found in [2–5].

Let L^A be the matrix whose rows are the first (in our ordering) vectors

$$H_A(t_1), ..., H_A(t_m)$$

which are linearly independent and any other vector $H_A(t)$ is a linear combination of them.

A is said to be *reduced* (respectively *minimal*) if no two columns of L^A are identical (respectively no column of L^A is a positive combination of the others).

The behavior set of a positive tree automaton $A=(\Sigma, Q, \alpha)$ is

$$B(A) = \{A_\zeta / \zeta \in \mathbb{R}_+^{n \times 1}, n = \text{card } Q\},$$

where the function $A_\zeta : T_\Sigma \to \mathbb{R}_+$ is given by

$$A_\zeta(t) = H_A(t) \cdot \zeta, \qquad t \in T_\Sigma.$$

$A_\zeta(t)$ is the positivity by which the automaton A (with final positive distribution ζ) consumes the tree $t \in T_\Sigma$. A covers A' ($A \geqslant A'$) whenever $B(A) \supseteq B(A')$ and A is equivalent to A' whenever $B(A) = B(A')$.

Finally, A is state equivalent to A' iff $Be(A) = Be(A')$, where

$$Be(A) = \{A_{e_1}, ..., A_{e_n}\}$$

$e_1, ..., e_n \in \mathbb{R}_+^{n \times 1}$ being the standard positive distributions.

State equivalence is a classification method which is weaker than isomorphism and stronger than equivalence:

$$\begin{tikzcd} & \text{isomorphism} \arrow[dl, Rightarrow] \arrow[dr, Rightarrow] & \\ \text{equivalence} & & \text{state equivalence} \arrow[ll, Rightarrow] \end{tikzcd}$$

To each PTA $A = (\Sigma, Q, \alpha)$ a positive tree representation $\Delta(A) = (\Sigma, Q, y_A, \varphi_A)$ can inductively be attached in the following manner:

— $\varphi_A(x) = I_n$ (the unit $n \times n$ matrix)

— for every tree $\tau \in P_\Sigma$ of the form

$$\tau = \sigma(t_1, ..., t_{i-1}, x, t_{i+1}, ..., t_n) \qquad (+)$$

and every pair of states $q, p \in Q$

$$\varphi_A(\tau)_{qp} = \bar{\alpha}_\sigma(H_A(t_1), ..., H_A(t_{i-1}), q, H_A(t_{i+1}), ..., H_A(t_n))(p)$$

— for an arbitrary $\tau \in P_\Sigma$, $\tau = \tau_1 \cdots \tau_\kappa$ (all τ_j being of the form $(+)$ above)

$$\varphi_A(\tau) = \varphi_A(\tau_1) \cdots \varphi_A(\tau_\kappa)$$

— for every $c \in \Sigma_o$, $\varphi_A(c) = \alpha_c$.

The coherence condition $(*)$ of Section 2 is immediately verified because it holds

$$y_A(c)\, \varphi_A(\tau) = H_A(t) \qquad \text{for all} \quad t = c\tau \quad (c \in \Sigma_o, \tau \in P_\Sigma).$$

Obviously $L^A = L^{\Delta(A)}$.

Example 5. Take the ranked alphabet Σ with $\Sigma_2 = \{\sigma\}$ and $\Sigma_o = \{c\}$ and consider the bottom up PTA A whose state set is $Q = \{1, 2, 3, 4\}$ and move functions

$$\alpha_\sigma \colon Q^2 \to \mathbb{R}_+^Q \qquad \alpha_c \colon Q \to \mathbb{R}_+$$

defined by

$$\begin{aligned}\alpha_\sigma(i, j) &= (\tfrac{1}{2} \quad 0 \quad \tfrac{1}{2} \quad 0), &&\text{if} \quad i, j \in \{1, 2\}\\ &= (0 \quad \tfrac{1}{2} \quad 0 \quad \tfrac{1}{2}), &&\text{if} \quad i, j \in \{3, 4\}\\ &= (0 \quad 0 \quad 0 \quad 0), &&\text{else}\end{aligned}$$

and

$$\alpha_c = (1 \quad 1 \quad 1 \quad 1).$$

An easy induction argument shows that its reachability map $H_A\colon T_\Sigma \to \mathbb{R}_+^Q$ is given by

$$H_A(t) = 2^{\text{size}_\sigma(t)}(1 \quad 1 \quad 1 \quad 1), \qquad \forall t \in T_\Sigma.$$

The associated PTR $\Delta(A)$ is now defined by the formulas

$$\varphi_A\left(\begin{smallmatrix} & \sigma & \\ \diagup & & \diagdown \\ t & & x\end{smallmatrix}\right) = 2^{\text{size}_\sigma(t)}\begin{pmatrix}1 & 0 & 1 & 0\\ 1 & 0 & 1 & 0\\ 0 & 1 & 0 & 1\\ 0 & 1 & 0 & 1\end{pmatrix} = \varphi_A\left(\begin{smallmatrix} & \sigma & \\ \diagup & & \diagdown \\ x & & t\end{smallmatrix}\right) \qquad t \in T_\Sigma$$

$$y_A(c) = \alpha_c.$$

The results of the previous section can now be translated as follows:

THEOREM 4. *Let A, A' be two equivalent PTAs with n and n' states, respectively. Then*

(i) $\text{rank}(L^A) = \text{rank}(L^{A'})$

(ii) $L^{A'} = L^A \cdot P$ *and* $L^A = L^{A'} \cdot P'$ *for appropriate positive matrices* P *and* P' *and*

(iii) $\text{ph}\{\omega_1^A, ..., \omega_n^A\} = \text{ph}\{\omega_1^{A'}, ..., \omega_{n'}^{A'}\}$,

where ω_i^A *(respectively* $\omega_i^{A'}$*) denotes the ith column of* L^A *(respectively* $L^{A'}$*).*

PROPOSITION 5. *For state equivalent PTAs A, A' with n, n' states respectively, it holds*

$$\{\omega_1^A, ..., \omega_n^A\} = \{\omega_1^{A'}, ..., \omega_{n'}^{A'}\}.$$

If, further, A and A' are reduced, then $n = n'$ and the columns of L^A are a permutation of the columns of $L^{A'}$.

THEOREM 5. *Let A and A' be two PTAs. The conditions below are equivalent*:

(i) $A \geqslant A'$.

(ii) *There exists a positive matrix P such that*

$$\begin{cases} y_A(c) \cdot P = y_{A'}(c), & \text{for all} \quad c \in \Sigma_o \\ L^A \cdot \varphi_A(\tau) \cdot P = L^A \cdot P \cdot \varphi_{A'}(\tau), & \text{for all} \quad \tau \in P_n \end{cases}.$$

Proof. Apply Theorem 2(iv) to $\Delta(A)$ and $\Delta(A')$.

Consider two PTAs A and A' and assume that Q is a positive matrix solution of the equation

$$L^A \cdot Q = J(\Delta(A), \Delta(A')),$$

where the matrix $J(\Delta, \Delta')$ has been defined in the proof of Proposition 2. Then

PROPOSITION 6. $A \geqslant A'$ *iff*

$$\begin{cases} y_A(c) \cdot Q = y_{A'}(c), & \text{for all} \quad c \in \Sigma_o \\ L^A \cdot \varphi_A(\tau) \cdot Q = L^A \cdot Q \cdot \varphi_{A'}(\tau), & \text{for all} \quad \tau \in P_n \end{cases}.$$

Proof. Let P be the positive matrix granted by the previous theorem; it holds

$$L^A \cdot P = J(\Delta(A), \Delta(A'))$$

so that

$$L^A \cdot Q = L^A \cdot P.$$

Take, next, an arbitrary tree $t \in T_\Sigma$ and decompose $H_A(t)$ along the rows $H_A(t_1), ..., H_A(t_m)$ of L^A

$$H_A(t) = \sum_j \alpha_j H_A(t_j) \qquad \alpha_j \in \mathbb{R}.$$

Then

$$H_A(t) \cdot Q = \sum_j \alpha_j H_A(t_j)\, Q = \sum_j \alpha_j H_A(t_j)\, P = H_A(t)\, P = H_{A'}(t).$$

Following arguments of Theorem 2 we can show that Q satisfies the proposed equations. For the converse we again proceed as in Theorem 2 by replacing P with Q. ∎

THEOREM 6. *We can decide whether or not $A \geqslant A'$.*

Proof. The matrix L^A (and thus $J(\Delta(A), \Delta(A'))$) can algorithmically be determined because its rows are the first (in our ordering) linearly independent vectors generating the finite list $H_A(t)$, height$(t) \leqslant n$, where n is the number of states of A. On the other hand, using linear programming (cf. [9]) we can decide whether or not the equation

$$L^A \cdot X = J(\Delta(A), \Delta(A')) \tag{E}$$

has a solution Q which is a positive matrix. Finally, for such a matrix Q we check whether the conditions of Proposition 6 are valid or not. ∎

COROLLARY. *Equivalence of PTAs is decidable.*

When dealing with tree automata A, A' having costs on the semiring $\mathbb{N}$, then Theorem 6 and its corollary still remain true. Indeed, the entries of the matrices

$$L^A \qquad \text{and} \qquad J(\Delta(A), \Delta(A'))$$

are all nonnegative integers, while Eq. (E) has an effectively determined solution set as confirms Theorem 3.9, [6].

ACKNOWLEDGMENT

I express my gratitude to the referee for his fruitful comments and suggestions.

Received May 30, 1995; final manuscript received July 8, 1997

REFERENCES

1. Berstel, J., and Reutenauer, C. (1982), Recognizable formal power series on trees, *Theoret. Comput. Sci.* **18**, 215–248.
2. Bozapalidis, S. Coalgebras and formal power series on trees, Submitted for publication.
3. Bozapalidis, S. (1994), Representable tree series, *Fund. Inform.* **21**, 367–389.
4. Bozapalidis, S. (1991), Effective construction of the syntactic algebra of a recognizable series on trees, *Acta Inform.* **18**, 251–263.
5. Bozapalidis, S., and Alexandrakis, A. (1989), Répresentation matricielles des séries d'arbres reconnaissables RAIRO, *Inform. Theor. Appl.* **23**, 439–449.
6. Eilenberg, S. (1974), "Automata, Languages and Machines," Vol. A, Academic Press, New York.
7. Eilenberg, S. (1977), "Automata, Languages and Machines," Vol. B, Academic Press, New York.
8. Gass, S. I. (1985), "Linear Programming," 5th ed., McGraw–Hill, New York.
9. Gecseg, F., and Steinby, M. (1984), "Tree Automata," Akademiai Kiado, Budapest.
10. Paz, A. (1971), "Probabilistic Automata," Academic Press, New York.
11. Reutenauer, S. (1989), "Aspects Mathématiques des riseaux de Petri," Masson, Paris.
12. Salomaa, A., and Soittola, M. (1978), "Automata Theoretic Aspects of Formal Power Series," Springer-Verlag, Berlin/New York.